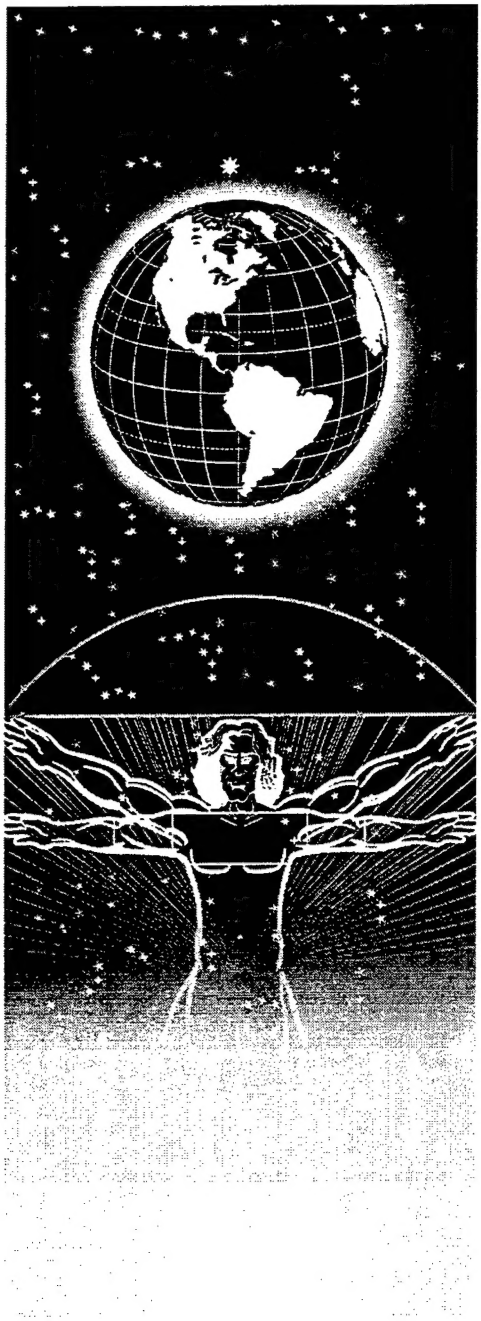
AL/HR-TP-1997-0024



UNITED STATES AIR FORCE RESEARCH LABORATORY

SYSTEM ARCHITECTURE FOR A DESKTOP DECISION TRAINER

Brian Van de Wetering

SYSTEMS ENGINEERING ASSOCIATES
2851 Camino Del Rio South
Suite 410
San Diego, CA 92108-3843

Sharon K. Garcia

HUMAN EFFECTIVENESS DIRECTORATE
MISSION CRITICAL SKILLS DIVISION
7909 Lindbergh Drive
Brooks AFB, TX 78235-5352

December 1998

NOTICES

This report is published in the interest of scientific and technical information exchange and does not constitute approval or disapproval of its ideas or findings.

Using Government drawings, specifications, or other data included in this document for any purpose other than Government-related procurement does not in any way obligate the US Government. The fact that the Government formulated or supplied the drawings, specifications, or other data, does not license the holder or any other person or corporation, or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

The Office of Public Affairs has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This report has been reviewed and is approved for publication.

SHARON K. GARCIA, Ph.D.
Project Scientist

R. BRUCE GOULD, Ph.D.
Acting Chief
Mission Critical Skills Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1998	Final - February 1992 - February 1997

4. TITLE AND SUBTITLE

System Architecture for a Desktop Decision Trainer

5. FUNDING NUMBERS

C - F33615-91-C-0007
PE - 62205F
PR - 1123
TA - C2
WU - 76

6. AUTHOR(S)

Brian Van de Wetering
Sharon K. Garcia

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Systems Engineering Associates
2851 Camino Del Rio South, Suite 410
San Diego, CA 92108-3843

8. PERFORMING ORGANIZATION

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory
Human Effectiveness Directorate
Mission Critical Skills Division
7909 Lindbergh Drive
Brooks Air Force Base, TX 78235-5352

10. SPONSORING/MONITORING

AL/HR-TP-1997-0024

11. SUPPLEMENTARY NOTES

Air Force Research Laboratory Technical Monitor: Dr. Sharon K. Garcia; (210) 536-3989

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release ;distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This paper describes the software design of the Desktop Decision Training System (DDT). It documents how the system's software implements an instructional strategy to train complex decision-making skills to Logistics Command and Control (LC2) personnel. This paper is intended to be a pragmatic guide for the DDT's software programmers and technical management.

14. SUBJECT TERMS

Computer-Based Training
Decision Making
Instructional Strategy
Logistics Command and Control (LC2)

15. NUMBER OF PAGES
18

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

CONTENTS

PREFACE

This paper describes the software design of the Desktop Decision Trainer (DDT) developed in support of the "Desktop Training for Logistics Command and Control (LC2)" research and development effort. This project was accomplished under contract number F33615-91-C-0007, with System Engineering Associates (SEA), San Diego, CA. Management of this project was provided by the Human Resources Directorate, Technical Training Research Division, Instructional Systems Branch (AL/HRTD).

SUMMARY

Logistics Command and Control (LC^2) units must ensure that core and augmentee personnel are fully trained in the combat critical skills of decision making. At present, existing training capabilities are inadequate. They consist of primarily expensive and manpower-intensive exercises which afford only sporadic training opportunities. These opportunities are considered insufficient to achieve and maintain the skill levels required for successful combat operations. The need for more accessible, affordable and less manpower-intensive training continues to exist.

In 1991, the Armstrong Laboratory Human Resources Directorate (AL/HR) began research and development efforts to improve decision-making training for Logistics centers throughout the Air Force. A contract was let with Systems Engineering Associates (SEA) to produce a computer-based decision trainer to provide individual instruction and enable students to practice solving realistic logistics problems within a simulation environment. The project began in February 1992 and concluded in February 1997.

This paper describes the software design of the decision trainer, and in particular, how the system implements the instructional strategy developed to teach decision-making

1. DDT ARCHITECTURE REVIEW

User feedback from the first version of the DDT indicated that although interactive story model was very useful, the user interface needed improvement and the logistical model was more complex than was needed to teach the skills in question. In order to quickly respond to user feedback about the interface and to lower development cost, the architecture of the DDT has been changed to include Visual Basic for creating the user interface, while still preserving the investment in the C++ code used to implement the interactive story. This document describes the methodology used to replace the C++ user interface code in the original DDT with Visual Basic. The interface code and the logistic simulation code were stripped from the original DDT and the remaining interactive story code is compiled as a DLL that to interface with Visual Basic. This DLL which exports both custom controls and DLL functions to Visual Basic. The following figure illustrates the coupling between the DDT simulation and Visual Basic.

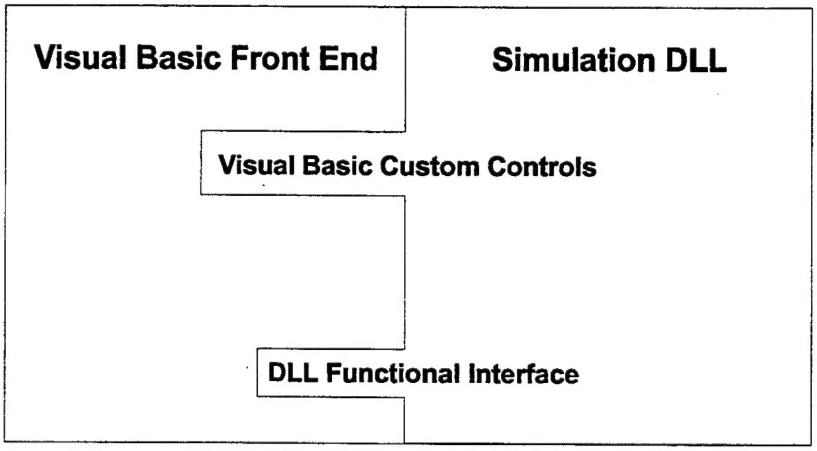


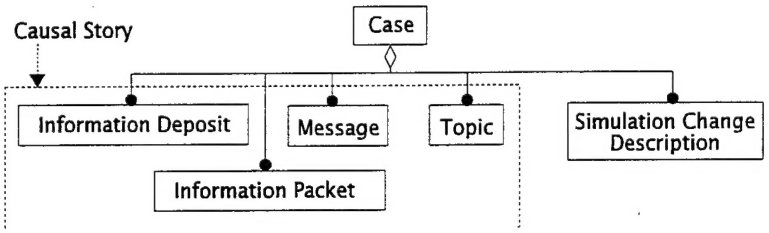
Figure 1. Coupling between DDT Simulation and Visual Basic

The original DDT simulation is divided into three engines. The first engine models a logistical network, the second models an interactive story, and the third models a decision option structure. The new DDT simulation DLL contains only the interactive story model and the agency object from the logistical network model. The logistical network model and the decision option structure have been simplified and implemented in Visual Basic using MS Access databases. MS Access databases have also been used to implement portions of the interactive story. The following section describes the DDT simulation DLL's interactive story engine and its interfaces with Visual Basic. Subsequent sections discuss the databases used to implement the logistics network and the decision options structure.

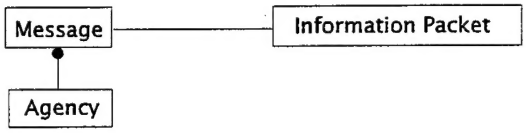
2. Interactive Story Engine

The interactive story engine is managed by the case objects. Each case object consists of a list of messages, topics, and information deposits. Below is a list of the objects used in the story engine and a diagram of each object's class structure.

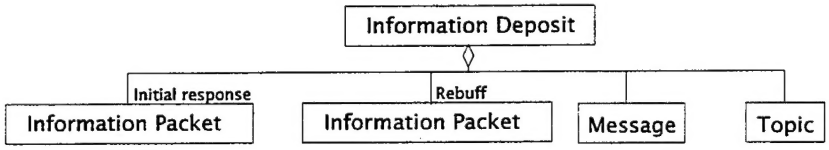
2.1 Case Object:



2.2 Message:



2.3 Information Deposit:



In the original DDT, these objects contain C++ user interface code within their structure. The two main user interface hooks for retrieving information from the story engine are the message desk and agency query screens. Below is a description of the original message desk and agency query screens, along with their Visual Basic versions.

2.4 Original Message Desk GUI

The original message desk screen is controlled by the DeskManager object. This object manages five lists of messages.

- TodoDesk - List of messages on the todo list window.
- FileDesk - List of messages on the file cabinet window.
- TrashDesk - List of messages placed in the trash can.
- ForwardDesk - List of forwarded messages.
- MessageDesk - List of message on the message desk window.

The DeskManager object handles the display and movement of messages in the GUI using the following four methods.

- AddToList - Adds a message to one of the five lists.
- RemoveFromList - Deletes a message from one of the five lists.
- HideMessageList - Hides the message for the give list.
- ShowMessageList - Shows the message for the given lists.

2.5 Visual Basic Message Desk GUI

To implement the message desk user interface in Visual Basic, the engine exports the following group of custom controls, event methods, and DLL functions:

2.5.1 MessageList Custom Control

The MessageList custom control exports the four lists used by the DeskManager object to VB. This Control acts as a data server from the simulation DLL to VB. The control is not visible to the user.

2.5.2. Properties:

LpMessageObj ——	Long pointer to a message object.
MessageCount ——	Count of objects on the MessageDeskList.
MessageIndex ——	Index of current list object.
LpFolderObj ——	Long pointer to a message object.
FolderCount ——	Count of objects on the FileDeskList.
FolderIndex ——	Index of current list object.
LpForwardObj ——	Long pointer to a message object.
ForwardCount ——	Count of objects on the ForwardDeskList.
ForwardIndex ——	Index of current list object.
LpTrashObj ——	Long pointer to a message object.
TrashCount ——	Count of objects on the TrashDeskList.
TrashIndex ——	Index of current list object.

2.5.3. Events:

MessageAdded(List ID, LpMessageObj)

MessageDeleted(List ID, LpMessageObj)

These events are triggered by changes to the message lists listed above. The MessageAdded event is triggered when mail arrives. VB code for these events handles displaying messages on the desk and folder windows.

2.5.4. DLL Functions:

MoveMessage(SourceListID, DestinationListID, LpMessage)

Tells the engine to move a message from one list to another. The engine then fires the AddMessage and DeleteMessage events.

2.6 Message Custom Control

The Message custom control displays a picture of a message. The control sits on the message desk window. Visual Basic (VB) code instantiates message controls depending on the contents of the message lists. When one of these controls is clicked on, it calls the Show method of its InfoPac causing the available topics list to be reevaluated, then it passs the database key of its InfoPac to VB in the click event so that VB can display the contents of the InfoPac.

2.6.1. Properties:

LpMessageObj ——	Long pointer to a message object.
Subject ——	String subject of message.
Sender ——	String Name of sending agency.
LpInfoPac ——	Long pointer to infopac object.
Picture ——	Bitmap or metafile depiction of the closed message.

2.6.2. Events:

Click(FileName, FileType)

2.7. Original Agency Query User Interface

The original agency query user interface consists of an agency listbox and a topic listbox. The user selects an agency from the agency listbox then selects a topic from the topic listbox. The listboxes contain pointers to agency and topic C++ objects. Once the user has selected an agency and topic, DoInfoDeposit is called to display the information packet and mark the topic as discovered.

2.8. Visual Basic Agency Query User Interface

To implement the agency query user interface in Visual Basic, the simulation exports the following group of custom controls.

2.9. AgencyList Custom Control (Non-Location)

The AgencyList custom control exports the list of non-location agencies, and optionally displays them in a list box for user selection. The LpAgencyObj, AgencyIndex, AgencyName, CommMug properties are all set when the user clicks to reflect the currently selected agency in the list box. The currently selected agency in the list box changes whenever the LpAgencyObj or AgencyIndex properties are set by VB code. A click event is generated whenever the user selects an agency in the listbox, allowing the control to be used both as a VB interface to the agencies list and as an agency selection control.

2.9.1. Properties:

LpAgencyObj ——	Long pointer to a agency object. (read-write at runtime)
AgencyCount ——	Count of agencies on the list. (read-only at runtime)
AgencyIndex ——	Index of current list object. (read-write at runtime)
AgencyName ——	String name of the current agency. (read-only at runtime)
CommMug ——	Picture property. (read-only at runtime)
Visible ——	Determines if the control displays a listbox.

2.9.2. Events:

Click()

2.10. TopicList Custom Control

The TopicList custom control exports the list of currently active topics, and optionally displays them in a list box for user selection. The LpTopicObj, TopicIndex, TopicIndex properties are all set when the user clicks to reflect the currently selected topic in the list box. The currently selected topic in the list box changes whenever the LpTopicObj or TopicIndex properties are set by VB code. A click event is generated whenever the user selects a topic in the listbox, allowing the control to be used both as a VB interface to the active topics list and as a topic selection control. The addition of the LpAgencyObj property allows the control to operate as a complete agency query control. When the LpAgencyObj property points to a valid agency, and the user choses a topic from the listbox, the control finds the appropriate InfoPac and returns the file name or database key and file type in the click event so that VB code can display the result of the agency query. The control also handles the topic list reevaluation.

2.10.1. Properties:

LpTopicObj,	Long pointer to a topic object.
TopicCount,	Count of topics on the list.
TopicIndex,	Index of current list object.
TopicIndex,	String name of topic.
LpAgencyObj,	Long pointer to an agency object. (read-write at runtime)
QueryOnClick,	Boolean indicating whether the control should actually perform the query when the user clicks.
QueryTrigger,	Boolean that when set to TRUE performs the query for the currently selected topic and the current value of LpAgencyObj, and programmatically generates the appropriate click event.

2.10.2. Events:

Click(FileName, FileType)

Place VB code in the Click event to "display" the contents of the file or database record.

3. Visual Basic Decision Options Structure

The Visual Basic decision options structure is implemented in a module called ***options.bas.*** This module contains the data structure definitions and the code for manipulating those data structures. Because the decisions options structure and the logistics network are so closely related, this module also contains the data structures and code for manipulating the logistics network. The logistics network configuration for each case is stored in an MS Access database. The decision options structure consists of the following Visual Basic data structures:

```
Type Leg
    EmbarkPortIdx As Integer
    ArrivalPortIdx As Integer
    FMIdx As FreightMission
End Type

Type Shipment
    SourceIdx As Integer
    Quantity As Integer
    DestIdx As Integer
    Legs(MAXLEGS) As Leg
    TotCost As Single
    Cost As String
```

```
    Risk As Single
    FirstArrival As String
    arrivalamount As String
End Type

Type DDTOption
    Shipments(3) As Shipment
    Amounts(13) As Single
    TempFMTableName As String
    Evaluated As Integer
    accessed As Integer
End Type
```

The student can respond to a logistics problem by specifying several (up to four) options and then choosing the best option as the solution to the problem. An option consists of several shipments (up to three) and each shipment consists of several legs. On each leg, the shipment travels via a particular Freight Mission. The decision options structure is built using the above types as the user interacts with the system. The commodities and freight missions that the user has to choose from are specified for each case in the case database. The case database consists of the following tables:

CaseParameters
Commodities
FMMaster
InfoPackets

The CaseParameters table contains basic information about the case including the name of the text file that defines the structure of the interactive story. This table also contains information used to evaluate the students solution to the case. The commodities table contains information about the amount of the commodity in question available at each location. The FMMaster table contains information about all of the scheduled freight missions available in the case. The InfoPackets table contains all of the text related to the interactive story. In the future these fields might also contain multimedia data such as sound or video.

4. Reference Desk

The Reference desk can be accessed by the student during the simulation to provide basic information about commodities, locations, etc. In the original DDT this feature was not yet implemented. The current interface has two levels of menus with the actual information being presented in fixed panes on the screen. This is accomplished using data aware widgets in the fixed panes to display pictures contained in an MS Access database, and using the menu widgets to position the database.